The Workshops of the Thirty-Second
AAAI Conference on Artificial Intelligence

A Framework for Utilizing Lab Test Results for Clinical Prediction of ICU Patients

Mohammad M. Masud, Muhsin Cheratta
College of Information Technology
United Arab Emirates University
PO Box 15551 Al Ain, UAE
{m.masud, muhsin.j}@uaeu.ac.ae

Abstract

Clinical decision support has gained significant attention in recent years, especially with the advancement of data analytics techniques. One active research area in this domain is survival prediction or deterioration prediction of critical care patients, such as intensive care unit (ICU) patients. Usually, ICUs are equipped with continuous monitoring devices, which monitor vital signs such as heart rate, blood pressure, Oxygen saturation and so on. In addition to this, ICU patients also undergo different pathological (i.e., lab) tests. Recent studies claim that vital signs can be used to predict the near future status of a patient, with the help of predictive analytics. However, in this work, we investigate the usefulness of lab test results in patient survival prediction, which have been rarely used for this purpose. We propose a framework for utilizing the lab test data for this clinical prediction task. We encounter several challenges associated with this task, including variable-length feature vector, longitudinal features, missing data, class imbalance and high dimensionality. The proposed work addresses most of these challenges under this single framework. In this framework we propose a novel orthogonal clustering technique to reduce data dimensions as well as missing data. We also propose a systematic approach to inject informative background knowledge into the data and increase the prediction performance. The proposed technique has been evaluated on a real ICU patients database, achieving notable success in reducing 66% of the data dimensions without discarding any feature, while improving the weighted average F1-score 5% on average and achiving about 3 times speedup. We believe that the proposed technique will provide a powerful framework in the field of clinical and healthcare data analytics and healthcare decision support.

1 Introduction

Research in clinical prediction and decision support has gained increasing attention in recent years with the proliferation of digital healthcare data, advancement of data analytics techniques, and availability of high performance computing services. There are many different sources and forms of healthcare data, such as electronic health records (EHR), medical imaging, medical text data (e.g. nurse notes), and public health data. In this work, we are interested mainly on utilizing EHR data of intensive care unit (ICU) patients for survival prediction of ICU patients. ICU patients are kept under close monitoring and testing using sophisticated devices. This monitoring can be continuous (e.g. vital signs), or intermittent, such as the lab tests results. In addition to these, there are records of medication, nurse notes, demographic data as well as administrative (e.g. admissions) and procedural (e.g. caregiver name) information.

Recent research demonstrates the possibilities of utilizing some of these data, such as vital signs, to predict patient situation (e.g. deterioration) ahead of time and warn caregivers to take timely and reliable measures to save the patient's life. Therefore, the main goal in this direction of research is to construct an automated early warning system based on reliable prediction. There are many challenges involved in building such a system, some of which are mentioned in the literature ((Mao et al. ; Fialho et al. 2012; Baumgartner, Rdel, and Knoll 2012; Cheng et al. 2013)). Our goal is also aligned with this goal, i.e., construction of an early warning system. However, instead of vital signs, we focus on using the laboratory test results, which have been rarely used for survival prediction. We would like to investigate the challenges and prospects of using test results for survival prediction prediction and propose effective solutions that would complement the findings in the literature that utilizes vital signs.

There are several challenges that need to be addressed to utilize the lab test data by considering each lab test as a feature. The first challenge is a variable length feature vector, which occurs because each patient undergoes a different subset (possibly overlapping) of lab tests. Therefore, there will be no uniform feature vector across all patients. However, most learning algorithms require uniform feature vector, which forces us to introduce missing values into the feature vector of each patient. Second, the data suffers from high dimensionality, which occurs because of the large number different possible tests that can be done on a patient. Third, class imbalance is observed, as is observed in many medical domains. Finally, the lab tests are longitudinal features, i.e., the same test may be done more than once on a patient. Therefore, the challenge is how to effectively and efficiently handle the multiple values of the same feature.

In this work, we target the first three challenges, i.e., missing values, high dimensionality, and class imbalance. The first two challenges are interrelated, because the missing val-

ues are introduced mainly because of high dimensionality, and they bring noise, redundancy, and sparsity in the dataset. One obvious solution is to apply some kind of feature selection. However, because of the sparsity and missing values, traditional feature selection techniques do not perform well. This situation is aggravated by the class imbalance problem. We propose a novel framework based on orthogonal clustering of the data, which involves vertical clustering (i.e., feature clustering) followed by horizontal clustering (i.e., patient clustering) that significantly reduce the feature dimensions and noise in the dataset. After the clustering, we apply missing value replacement and class balancing on each cluster, and train a classification model from each cluster. Then the ensemble of models is used to classify new patient data.

Our contributions are as follows. First, we propose an orthogonal clustering (vertical + horizontal) approach to reduce data dimension as well as missing data without losing any feature (i.e., performing any feature selection). To the best of our knowledge, this is the first work to propose such clustering for missing data reduction. Second, we propose a framework that systematically handles the missing data, high dimension, and class imbalance problem. Third, we demonstrate how carefully chosen background knowledge can be integrated into existing data in order to improve prediction. Fourth, we apply the proposed technique on a real patient dataset, and achieve notable improvement over state-of-the-art techniques in terms of prediction accuracy and running time. We believe our approach will be very useful in clinical prediction (e.g. survival or mortality) of ICU patients and a useful tool can be developed from it for to aid in clinical decision support.

The rest of the paper is organized as follows. Section 2 discusses the works relevant to our technique. Section 3 describes the proposed approach in details. Then Section 4 reports the experiment details and analyzes the results. Finally, Section 5 concludes with directions to future research.

2 Related work

There are mainly two broad categories of works related to the proposed one. The first category deals with different clinical and healthcare support aspects of ICU patients. The other category of related work applies machine learning in general for medical decision support.

First we discuss the works that specially deal with ICU patients. Mao et al. (Mao et al.), developed a data-mining approach to predict deterioration of patients in the ICU. They used time-series data obtained from different sensors attached to patient's body, such as blood pressure, heart rate, O2 saturation and so on. They preprocessed these time-series data and derived several features. Finally, they applied different feature selection and optimization techniques to build prediction model, which observes good prediction rate. Pirracchio et al. (Pirracchio et al. 2015) proposed a learning algorithm to predict mortality of ICU patients and successfully used in real hospitals. Cismondi et al. (Cismondi et al. 2013) addressed a different goal involved with ICU patients, namely, how to reduce unnecessary lab testing in the ICU. However, they only focus on gastrointestinal bleeding. In our work, we are targeting all cases in the ICUs.

Some other relevant work dealing with ICU patients are as follows. Fialho et al. (Fialho et al. 2012) proposed a feature selection technique to find the best features in order to predict ICU readmissions. Cheng et al. (Cheng et al. 2013) proposed a clinical decision support system using association rule mining that finds associations among various variables such as patients' conditions, length of ICU stays and so on, and reports interesting findings.

Our proposed work is different from the above in that most of the above works use vital signs or demographic variables, whereas we use only lab test results. Also, we study different background knowledge, carefully choose some of them that are most informative, and add them to the data, which most other works did not consider.

The other category of related work are all approaches that in general deal with data mining based solutions for developing clinical decision support systems. Herland et al. (Herland, Khoshgoftaar, and Wald 2013) did a comprehensive survey on this topic, i.e., clinical data mining applications on big data in health informatics. Celi et al. (Celi et al. 2011) applied a statistical approach to predict mortality among patients with acute kidney injury. Cai et al. (Cai et al. 2016) proposed a Bayesian network approach to develop models using EHR for real-time prediction of several targets, including length of hospital stay, mortality, and readmission of hospitalized patients.

Our proposed orthogonal clustering approach can be thought of a variant of bi-clustering (Pontes, Girldez, and Aguilar-Ruiz 2015). However, the main difference between the proposed clustering and bi-clustering is that the proposed clustering is done without using any feature value, but by considering missing data in computing the distance measure. Whereas other clustering techniques use the values of each feature to compute vector distance. Although our approach also applies data mining technique for survival prediction of ICU patients, it is more focused towards improving the prediction performance as well as efficiency in running time. Therefore, we believe it is applicable to any clinical decision support problem in general.

3 Proposed method

In this section first we give an overview of the proposed technique and then describe it in details.

3.1 High level description

Figure 1 shows the high level architecture of our approach. The ICU patients data are stored in a database. We use the lab test results data from the database, and combine it with some other tables such as demographic and administrative information. Then the data are cleaned and we choose only good quality data for training. Then we extract features from the processed data, where each feature corresponds to a lab test. Then we apply feature clustering followed by patient clustering on the feature set. This is followed by missing value replacement and class balancing process for each cluster. Then each cluster is used to build a prediction model using a classifier learning algorithm, thereby generating an ensemble of models. Finally, these ensembles are used to pre-

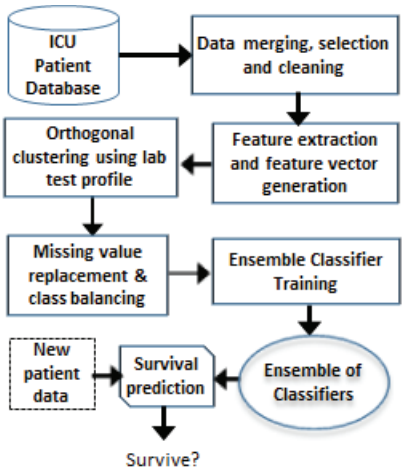


Figure 1: Architecture of the proposed approach

dict survival of new patients based on lab test results. Each of these steps are discussed below.

3.2 Database

The source of all datasets in this work is MIMIC-III (Physionet-MIMICIII). The database is collected by the Multi-parameter Intelligent Monitoring in Intensive Care (MIMIC) project at the Laboratory of Computational Physiology at MIT, funded by the National Institute of Biomedical Imaging and Bioengineering. The data was collected between 2001 to 2012 and contain more than 50,000 hospital admissions.

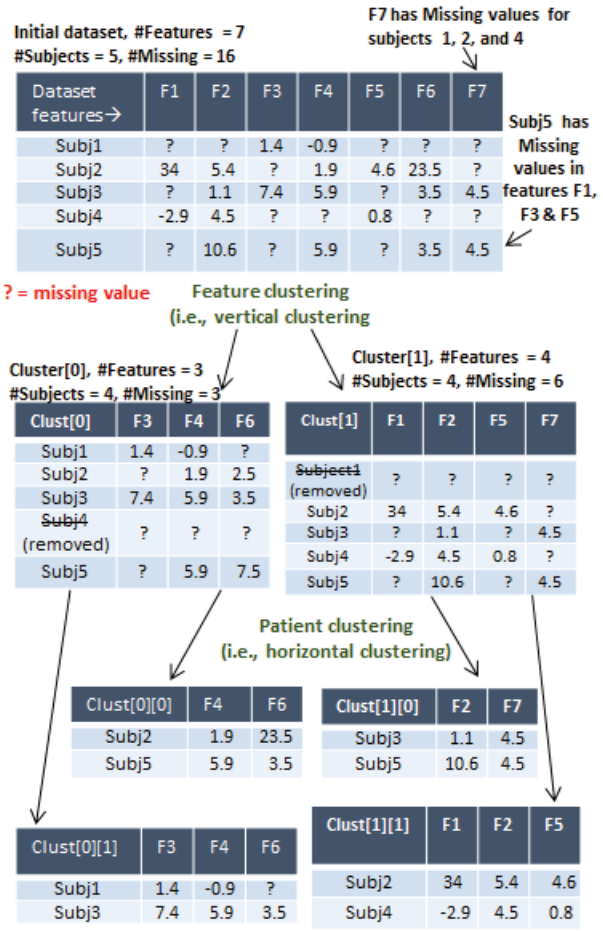
The database consists of 25 different tables, each containing different types of information, such as demographic data, hospital admission data, ICU stay data, medication information, lab test results, nurse notes and so on. Among these tables, we mainly use the lab test data (called *Labevents*) as well as some demographic and admission related information from other tables as discussed below.

The lab test data contains lab test result for each test done on each patient. For each lab test done, there is a record containing the numeric value of the result, a flag (binary) indicating whether the result was normal or abnormal, the unit of measurement, and date/time of the test performed.

We use the *Patient* table that contains contains the date of birth, gender, and whether the patient died or survived. We also use the *Admissions* table, which contains the date of admission, date of discharge, ethnicity, and diagnosis of the patient. We compute the age of a patient by joining the Patient table and Admissions table and deducting date of birth from the date of admission. Finally, we also use the *ICUStays* table that contains the length of stay of the patient.

3.3 Data merging, selection, and cleaning

We merge data from the Labevents table with the Patient table to assign class labels to each patient (i.e., dead/alive), which gives us a binary classification problem. In order to add background knowledge, we merge the corresponding



Dataset features→	F1	F2	F3	F4	F5	F6	F7
Subj1	?	?	1.4	-0.9	?	?	?
Subj2	34	5.4	?	1.9	4.6	23.5	?
Subj3	?	1.1	7.4	5.9	?	3.5	4.5
Subj4	-2.9	4.5	?	?	0.8	?	?
Subj5	?	10.6	?	5.9	?	3.5	4.5

Clust[0]	F3	F4	F6
Subj1	1.4	-0.9	?
Subj2	?	1.9	2.5
Subj3	7.4	5.9	3.5
~~Subj4~~ (removed)	?	?	?
Subj5	?	5.9	7.5

Clust[1]	F1	F2	F5	F7
~~Subject1~~ (removed)	?	?	?	?
Subj2	34	5.4	4.6	?
Subj3	?	1.1	?	4.5
Subj4	-2.9	4.5	0.8	?
Subj5	?	10.6	?	4.5

Clust[0][0]	F4	F6
Subj2	1.9	23.5
Subj5	5.9	3.5

Clust[1][0]	F2	F7
Subj3	1.1	4.5
Subj5	10.6	4.5

Clust[0][1]	F3	F4	F6
Subj1	1.4	-0.9	?
Subj3	7.4	5.9	3.5

Clust[1][1]	F1	F2	F5
Subj2	34	5.4	4.6
Subj4	-2.9	4.5	0.8

Figure 2: Illustration of orthogonal clustering

data from other tables (mentioned above). Data cleaning and selection are done as follows: First, we discard all lab test records that contain null or undefined values. Second, we choose only the patients with age 65 or more, as this group has the highest prevalence of ICU mortality. Third, we discard all patients who did not undergo any lab test. Finally, we only consider lab test records of the last ICU stay of a patient if he stayed more than once in ICU.

3.4 Feature extraction

Features are extracted from the Labevents table as follows. First we enumerate a list of all different lab tests done on the selected patients, which we will denote as the *feature set*. Then for each patient, we extract the test results for each feature from the records of that patient. This requires searching through several millions of lab test records. If the test was not administered on the patient, then the feature value is regarded as missing, and if the test was administered more than once, we take the last value. However, a more appropriate treatment for this case (i.e., more than one values) can be done by adapting a longitudinal feature treatment approach; but we vow to address this issue in future.

3.5 Orthogonal clustering (ORCU)

We divide the dataset into smaller clusters using a technique that we call *Orthogonal clustering* or ORCU, in short. This is a two stage clustering process, where in the first stage we vertically cluster the dataset by clustering the features and in the second stage, we horizontally partition the stage-1 clusters by clustering the subjects. The clustering is done based on the lab test profiles, that is, for the stage-1, we partition the dataset by grouping features such that each group of features (i.e., lab tests) are administered (approximately) on the same subset of patients. For the stage-2, we partition each stage-1 clusters by grouping the subjects such that each group is described by (approximately) the same subset of features (i.e., lab tests). Figure 2 illustrates the orthogonal clustering approach with a toy example.

Note that during the clustering we do not look into the feature values in the dataset; rather we only see whether a feature has a missing value in a feature vector or not. Both clustering processes reduce high dimensionality of data, as well as reduce sparsity. After stage-2 clustering, we perform missing value replacement, class balancing, and training a classification model from each cluster, which forms an ensemble of models. Missing value replacement on the clustered data is more useful than on the original data, as the proportion of missing values is much less in the latter. However, note that our main focus is not proposing a new approach for replacing missing value restore class balancing. Therefore, for both operations, we use the available techniques, such as replacing missing value with a mean or mode, and restore class balancing using SMOTE (Chawla et al. 2002).

In order to explain the clustering process, first we detail the notations and define some terms here.

Notations and definitions: Let $D_{M,N}$ be a dataset with M subjects, and N features, where $S = \{S_1, ..., S_M\}$ is the set of Subjects, i.e., patients; and $F = \{F_1, ..., F_M\}$ is the set of feature, i.e., lab tests. Therefore, the dataset $D_{M,N}$ is an MxN matrix, where each cell $d[i][j]$ consists of the feature value (i.e., test result) of the i-th patient S_i for the j-th feature F_j. Also, we denote each row and column of the matrix as follows: The i-th row, i.e., the feature vector for Subject S_i is

$D_{M,N}[i] = \{d[i][1], ..., d[i][N]\}$

And the j-th column, i.e., the column for Feature F_j is

$D^T_{M,N}[j] = \{d[1][j], ..., d[M][j]\}^T$

Definition 1 (Feature coverage vector $FV(S_i)$) *: The feature coverage vector $FV(S_i)$ of a Subject S_i is the set of features having non-missing values for this subject. Therefore, $FV(S_i) = \{F_{j_1}, ..., F_{j_q}\}$, where the feature $F_{j_k}, 1 \leq k \leq q$ has a non-missing value in the feature vector of Subject S_i (i.e., $d[i][j_k]$ is not missing value).*

In other words, $FV(S_i)$ represents the set of lab tests that were actually administered on the patient S_i. All other features have missing values in the feature vector of S_i.

Definition 2 (Subject coverage vector $SV(F_j)$) *: The subject coverage $SV(F_j)$ of a Feature F_j is the set of subjects having non-missing values for this feature in the feature vector. Therefore, $SV(F_j) = \{S_{i_1}, ..., S_{i_r}\}$, where the subject $S_{i_k}, 1 \leq k \leq r$ has a non-missing value for feature F_j in its feature vector (i.e., $d[i_k][j]$ is not missing value).*

In other words, $SV(F_j)$ represents the set of patients on whom the test F_j was actually administered. All other patients have missing values for F_j in their feature vector.

Definition 3 (Total missing count, $TMC(D_{M,N})$) *: The total missing count $TMC(D_{M,N})$ in the dataset is the total number of missing values in all the feature vectors. This can be expressed with the following equation:*

$$TMC(D_{M,N}) = \sum_{i=1}^{M}(N - |FV(S_i)|) \tag{1}$$

Now we define the distance between coverage vectors.

Definition 4 (Vector distance, $Dist(A, B)$) *: Let A and B be two coverage vectors, where both of them are feature coverage vectors or both of them are subject coverage vectors. The distance between A and B is the normalized vector dissimilarity, i.e., Jaccard distance:*

$$Dist(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \tag{2}$$

The value of which is between 0 (exactly the same) and 1 (most distant). The distance value indicates what percentage of the *combined* vector will have missing values when vectors A and B are combined. For example, if the distance is 0.7, it means 70% of the combined feature vector (i.e., $A \cup B$) will have missing values.

Vertical (stage-1) clustering by grouping features This is done by iterative binary clustering of the dataset. The binary clustering is done as follows. First, we randomly initialize two clusters, say C[0] and C[1], with the subject coverage of two randomly chosen features. Then for each remaining features f, we compute the distance between the subject coverage f and the clusters, and place f in the nearest cluster. After this placement, the subject coverage of the cluster is updated as necessary. Algorithm 1 sketches the binary clustering technique. Lines 1-3 of the algorithm initializes two clusters with randomly chosen features and corresponding subject vectors. Then the *for* loop in lines 4-17 considers each remaining feature, chooses the nearest cluster by distance (lines 10-16), and adds to the cluster.

Horizontal (stage-2) clustering by grouping subjects This is done by iterative binary clustering of each of the stage-1 clusters. The binary clustering is done as follows. First, we randomly initialize two clusters, say C[0] and C[1], with the feature coverage of two randomly chosen subjects. Then for each remaining subject s, we compute the distance between the feature coverage of s and the clusters, and place s in the nearest cluster. After this placement, the feature coverage of the cluster is updated as necessary. The algorithm can be obtained by a transformation of Algorithm 1 by using feature coverage instead of subject coverage, therefore, we omit the details.

Algorithm 1 Feature-Clustering

Input: $D_{M,N}$: Dataset to cluster
Output: $\{C[1], C[2]\}$ such that $C[1] \cup C[2] = D_{M,N}$ and $C[1] \cap C[2] = \phi$ and $TMC(C[1]) + TMC(C[2]) < TMC(D_{M,N})$

1: $Vi \leftarrow$ RandomSelect($SV(F_i)$) // Randomly choose a feature coverage vector
2: $Vj \leftarrow SV(F_j)$ such that $Dist(SV(F_i), SV(F_j))$ is the max
3: $C[1] = Vi, C[2] = Vj$
4: **for all** $SV(F_k) \in D_{M,N}$ **do**
5: **if** $F_k \in FV(C[1])$ or $F_k \in FV(C[2])$ **then**
6: continue //already in a cluster
7: **end if**
8: d1 = Dist ($SV(C[1]), SV(F_k)$) //equation 2
9: d2 = Dist ($SV(C[2]), SV(F_k)$) //equation 2
10: **if** $d1 < d2$ **then**
11: $FV(C[1]) \leftarrow FV(C[1]) \cup F_k$ //add to feature set
12: $SV(C[1]) \leftarrow SV(C[1]) \cup SV(F_k)$ //add to subject cov
13: **else**
14: $FV(C[2]) \leftarrow FV(C[2]) \cup F_k$ //add to feature set
15: $SV(C[2]) \leftarrow SV(C[2]) \cup SV(F_k)$ //add to subject cov
16: **end if**
17: **end for**

Algorithm 2 Orthogonal Clustering and Ensemble Training

Input: $D_{M,N}$: Training Data (feature matrix)
Output: Clusters of data $\{C[1], ...C[m]\}$ such that $\cup_{i=1}^{m}(C[i]) = D_{M,N}$
E: the ensemble classifier, $E = \{E_1, ...E_m\}$

1: $Q \leftarrow \phi$ //Queue
2: $X \leftarrow D_{M,N}$
3: $C \leftarrow \phi$
4: **if** splitCond(X, stage-1) **then**
5: $Q \Leftarrow (X$, stage-1) //Enque
6: **end if**
7: **while** $Q <> \phi$ **do**
8: $(X$, stg) $\Leftarrow Q$ //Deque
/* Perform vertical (stage-1) clustering (algorithm 1) or horizontal (stage-2) clustering */
9: **if** stg = stage-1 **then**
10: $B[1], B[2] \leftarrow$ Feature-Clustering(X) //vertical
11: **else**
12: $B[1], B[2] \leftarrow$ Subject-Clustering(X) //horizontal
13: **end if**
14: **for** $i \leftarrow 1$ to 2 **do**
15: **if** splitCond($B[i]$,stg) **then**
16: $Q \Leftarrow (B[i]$, stg) //Enque for further clustering
17: **else**
18: **if** stg = stage-1 **then**
19: $Q \Leftarrow (B[i]$, stg-2) //Enque for stage-2 clustering
20: **else**
21: $C \leftarrow C \cup B[i]$
/* No more splitting, add to the cluster list */
22: **end if**
23: **end if**
24: **end for**
25: **end while**
26: $m \leftarrow |C|$ //number of data groups
27: **for** $i \leftarrow 1$ to m **do**
28: $C[i] \leftarrow$ Replace-missingvalue($C[i]$)
29: $C[i] \leftarrow$ Class-balancing($C[i]$)
30: $E_i \leftarrow$ TrainClassifier($C[i]$)
31: **end for**

Algorithm 2 sketches the overall ORCU algorithm. First we check if the dataset should be split (line 4), based on an *split-condition*. The *condition* is that the cross validation accuracy of the dataset should be at least same as the parent dataset (i.e., the dataset from which we got this dataset). Since at the beginning, there is no parent dataset, we set the threshold to 50% (i.e., if cross validation acc $< 50\%$ then don't split). If the condition is true, the dataset is put into a queue (line 5) for further processing. The while loop between lines 7-25 repeatedly splits the data into smaller clusters (lines 9-13). When stage-1 splitting condition is not satisfied (lines 17-23), we try applying stage-2 splitting (line 19), otherwise, we add the cluster to the processed cluster list (line 21). Finally, each cluster goes through missing value replacement and class balancing (line 28-29), and we train a classification model from each cluster, which gives us an ensemble of models.

3.6 Ensemble classification

The ensemble of models $E = \{E_1, ..., E_m\}$ trained in the previous phase is used to classify new patients to predict the mortality. Algorithm 3 sketches the ensemble classification technique. We apply a weighted majority voting technique where the weight of each model is inversely proportional to the distance (feature coverage vector based) between the test instance and the corresponding dataset of the model. That is, lower distance has higher weight and vice-versa. This ensures that if there is a larger disparity in the feature sets, the weight will be smaller.

Algorithm 3 Classify

Input: x: Instance to classify, $E = \{E_1, ..., E_m\}$: Ensemble of classifiers
Output: y: The predicted class

1: **for** $i \leftarrow 1$ to m **do**
2: $y_i \leftarrow$ Classify(x, E_i) //Classify
3: $C[i] \leftarrow$ Dataset (i.e., cluster) for E_i
4: $u_i \leftarrow$ Dist($FV(x)$, $FV(C[i])$) //equation 2
5: **end for**
6: $U \leftarrow \min_{i=1}^{m} u_i$
7: **for** $i \leftarrow 1$ to m **do**
8: $w_i = U/u_i$ //weight calculation
9: **end for**
10: $W \leftarrow \sum_{i=1}^{m} w_i$
11: $y \leftarrow \sum_{i=1}^{m} w_i y_i / W$

4 Experiments

In this section we describe the experiments and analyze the results.

4.1 Data set and preprocessing

The source of all datasets in this experiment is MIMIC-III (Physionet-MIMICIII), as described in Section 3.2.

We first organize the patients into different demographic groups based on their age. Then we keep only the age group with 65 or more age, which exhibits the highest prevalence

of ICU mortality. The total number of patients our working dataset is 5,602. Among them, approximately 43% are labeled *alive*, and the rest are labeled *dead*.

4.2 Competing approaches:

Base: This is the Baseline, with no preprocessing applied.
BMR: Baseline with Missing value Replacement applied.
BSMC: Baseline with feature Selection, Missing value replacement and Class balancing applied. Feature selection is done using gain ratio criteria, and number of selected features is set equal to the average number of features in each cluster obtained by the proposed method (ORCUE).
ORCUE: Proposed ORthogonal ClUstering and Ensemble classification approach.

4.3 Parameters and other setup

Base classifiers: We use NaiveBayes (NB), Decision Tree (J48) Random Forest (RF), and Support Vector Machine (SVM) from the WEKA API (wek). For each base learner we use the default parameter settings available in WEKA. For missing value replacement, we use the WEKA attribute filter *ReplaceMissingValue*, which replaces the missing values with mean/median of the attribute. For class balancing we use the WEKA instance filter *SMOTE*, which generates synthetic data for the minority class. We generate the minority data such that minority/majority ratio becomes 45/55 in the dataset.

4.4 Evaluation

Evaluation metric: Unless mentioned otherwise, we use F1-measure:

$$F1 = 2.\frac{precision.recall}{precision + recall}. \quad (3)$$

We calculate the F1 for each class (i.e., "Alive", and "Dead") and then report the weighted average F1, as follows:

$$F_A = \frac{F1_{Alive} * Count_{Alive} + F1_{Dead} * Count_{Dead}}{Count_{Alive} + Count_{Dead}} \quad (4)$$

Evaluation data: We arrange the data in order of hospital admission time. Then we take the first two-thirds as the training data and the last one-third as the evaluation data, which are used for all competing approaches.

Performance comparison in terms of F_A : Figure 3 shows the performance of each competing approach in terms of F_A on the lab test data for each classifier. We observe that except NB, ORCUE exhibits the best performance. For example, for the RF classifier, the F_A score of ORCUE is 66%, whereas that of the nearest competitor BSMC is only 51%. Overall (averaging all classifier performance), ORCUE achieves 65%, which is 7% higher than the overall performance (58%) of the nearest competitors (Base and BSMC). The overall performance of BMR is the lowest (55.7%). This shows the disadvantage of missing value replacement on a dataset where there is a very high proportion of missing data.

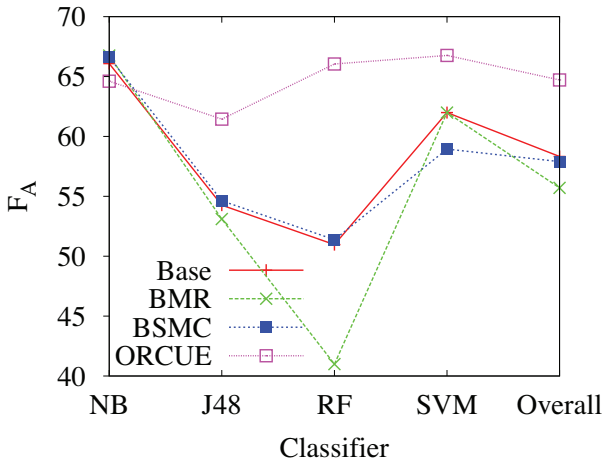


Figure 3: F_A on Lab test data

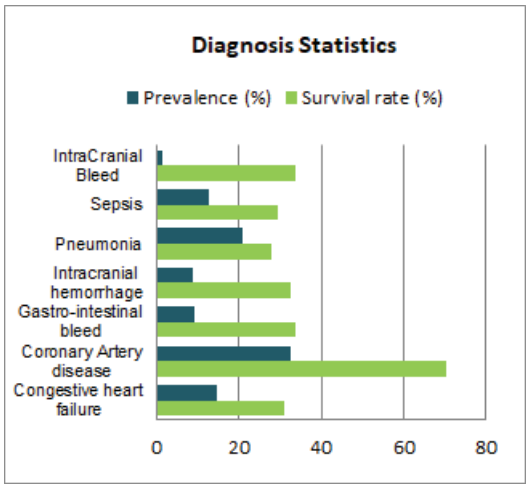


Figure 4: Prevalence and survival rates of top diagnosis for the patient in the dataset

Adding informative background knowledge: We study different background knowledge and carefully choose the best informative of them to the data. There are mainly two different categories of background knowledge added to the lab test features. First, we add the diagnosis information of each patient. ICU patients are diagnosed with different diseases and this information is available in the Admissions table. Statistics of prevalence rate (%) and survival rate (%) of the top diagnosis in our dataset are shown in Figure 4.

Among these diagnoses, we use the ones that exhibit relatively high predictive power; i.e., have high survival (or mortality) rate. For example, we choose Coronary Artery Disease as it has high survival rate (70%), as well as Pneumonia, that has high mortality rate (72%). We use each diagnosis as a binary feature, that is, if the patient is diagnosed with the corresponding diagnosis, then feature value is 1; otherwise it is 0. We will denote this data as ***Lab+Diag*** data.

The second category of background knowledge that we

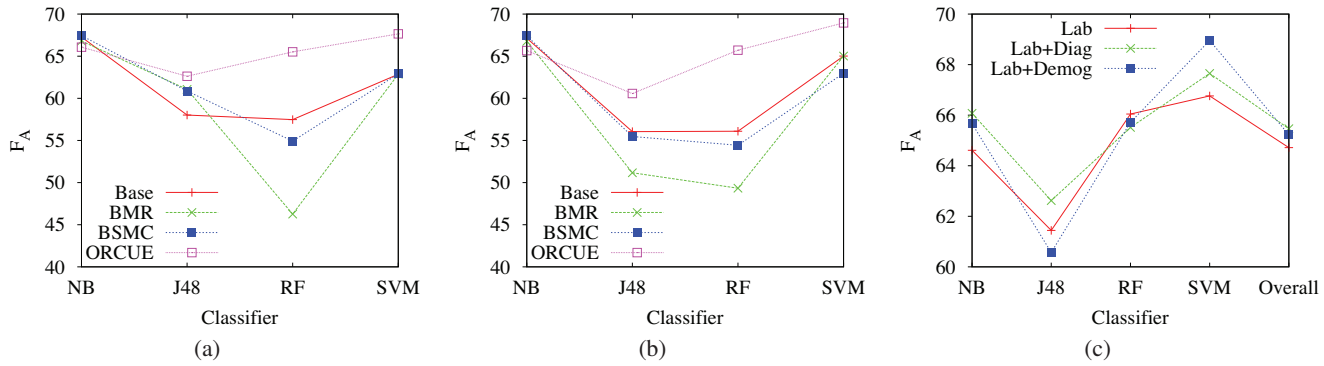


Figure 5: Performance in (a) Lab+Diag data and (b) Lab+Dmg data (c) Performance comparison in the three datasets for ORCUE

add to the lab tests are demographic and administrative data, namely, the *age* of the patient and length of stay (*los*) in the ICU. We will denote this data as ***Lab+Dmg*** data.

Figures 5(a,b,c) report the effect of adding selected informative background knowledge on prediction performance of different approaches. Figures 5(a) and (b) show the F_A values of the competing approaches when the lab test data are augmented with the Diagnosis data, and demographic data (age & length of stay), respectively. The relative performances of the competing approaches are similar to what we observe with only the lab test data, i.e., ORCUE still exhibits the best performance, followed by BSMC and Base. The advantage of adding the background knowledge becomes more evident with the Figure 5(c), which shows the relative performance of ORCUE on the three datasets (Labtest only, Lab+Diag, and Lab+Dmg). For NB and J48 classifiers, we observe about 2% improvement when Diagnosis data are added, whereas for SVM, about 1% improvement is observed after adding Diagnosis data. On the other hand, NB and SVM observers 1% and 2% improvements, respectively, when Demographic data are added. On average, we observe about 0.7% and 0.6% improvements in F_A when Diagnosis data and Demographic data are added, respectively. This indicates the effectiveness of careful selection and inclusion of background knowledge. We believe thoughtful inclusion of more background knowledge can further improve the prediction accuracies.

Table 1: Summary result on all datasets (average of all learners)

Competitor	Labtest	Lab+Diag	Lab+Dmg	Overall
Base	58.3	61.4	61.1	60.3
BMR	55.7	59.2	58.1	57.7
BSMC	57.9	61.5	60.1	59.8
ORCUE	**64.7**	**65.5**	**65.2**	**65.1**

Table 1 summarizes the findings above. Here we report the F_A of each approach for each dataset averaged over all the classifiers. We observe that ORCUE achieves the highest value for all datasets. On the labtest data, ORCUE is about

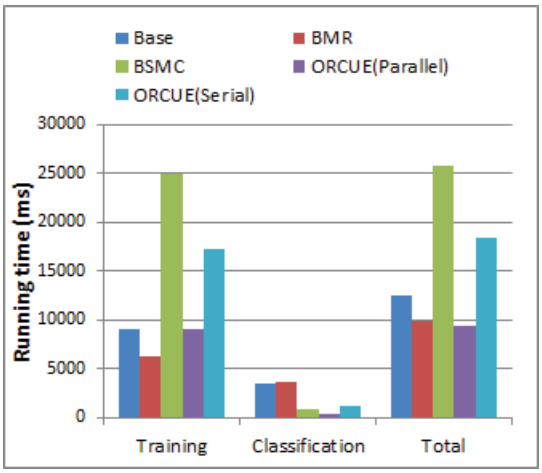


Figure 6: Average running time of each competing approach

6% higher than the nearest competitor, which is Base (64.7 vs 58.3). For the Lab+Diag and Lab+Dmg, we observe 4% higher value for ORCUE from the nearest competitor, and on average (shown in the overall column), ORCUE achieves 5% higher F_A than any competitor. This indicates the effectiveness of the proposed approach, which is mainly achieved because of the two-stage clustering by lab test profiles that reduces high dimensionality, noise, and sparsity in the data without losing much useful features.

Performance comparison in terms of running time: We report the running times, with a breakdown of training and classification times for each competing approach, averaged over all datasets and learning algorithms, in Figure 6. The training time for BMR includes missing value replacement time, that of BSMC includes feature selection, missing value replacement, and class balancing, and that of ORCUE includes clustering time, missing value replacement, and class balancing time. We report two variations of ORCUE, namely parallel and serial. The parallel version runs the stage-2 clustering, training of each cluster, and ensemble classification in parallel; whereas the serial version runs ev-

erything sequentially. The highest total running time is observed for BSMC (25.7 seconds), and the lowest for ORCUE parallel (9.4 seconds). Therefore ORCUE parallel is about 2.7 times faster than BSMC. ORCUE serial is also faster then BSMC. The high running time of BSMC occurs because of the feature selection, missing value replacement and class balancing processes. The main reason for ORCUE having lowest running time is that of efficient clustering as well as reduced dimension and noise. The total running times of other baselines (i.e., Base and BMR) are also higher than ORCUE parallel because of their high dimensionality and missing values.

Other statistics: Finally, we summarize the overall improvement of ORCUE over traditional predictive learning approach in table 2. We show different metrics such as the rate of data reduction due to ORCUE over original datasets, improvements in prediction accuracies and running times. These statistics are taken by averaging the results from all base learners. The overall improvement is noteworthy and an indication of the usefulness of the proposed technique.

Table 2: Overall improvements due to ORCUE

Criteria	Value
Total features before ORCU	624
Average features After ORCU	210
Data reduction by ORCU	66%
Missing value reduction by ORCU	45%
Overall improvement in prediction (compared to BSMC)	5%
Overall speedup achievement (compared to BSMC)	275%

5 Conclusion

We have proposed a framework for utilizing lab test data for ICU patient survival prediction with a novel orthogonal clustering technique. We also showed how to effectively integrate background knowledge into the dataset to improve prediction accuracy. The proposed technique has been proved to be very effective in reducing the data size, improving the prediction accuracy and significantly reducing the running time. We believe that the proposed work will be a valuable contribution to healthcare decision support, health data analytics as well as *Big data analytic* techniques where the data exhibits similar characteristics.

In future, we would like to address the issue of longitudinal feature that is observed with the lab tests. Furthermore, we would like increase the effectiveness of the prediction performance by combining the lab test data with the vital signs data. Finally, we would like to apply this technique to other areas of clinical decision support systems where the data exhibits similar properties.

Acknowledgment

This work was supported in part by UAEU Interdisciplinary Grant #: 31R060-Research Center- ZCHS-9-2014 .

References

Baumgartner, B.; Rdel, K.; and Knoll, A. 2012. A data mining approach to reduce the false alarm rate of patient monitors. In *IEEE Eng Med Biol Soc.*, 5935–8.

Cai, X.; Perez-Concha, O.; Coiera, E.; Martin-Sanchez, F.; Day, R.; Roffe, D.; and Gallego, B. 2016. Real-time prediction of mortality, readmission, and length of stay using electronic health record data. *Journal of the American Medical Informatics Association* 23(3):553–561.

Celi, L. A.; Tang, R. J.; Villarroel, M. C.; Davidzon, G. A.; Lester, W. T.; and Chueh, H. C. 2011. A Clinical Database-Driven Approach to Decision Support: Predicting Mortality Among Patients with Acute Kidney Injury. *J Healthc Eng* 2(1):97–110.

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; and Kegelmeyer, W. P. 2002. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.* 16(1):321–357.

Cheng, C. W.; Chanani, N.; Venugopalan, J.; Maher, K.; and Wang, M. D. 2013. Icuarm-an icu clinical decision support system using association rule mining. *IEEE J. of Translational Engg in Health and Medicine* 1:4400110–4400110.

Cismondi, F.; Celi, L.; Fialho, A.; Vieira, S.; Reti, S.; Sousa, J.; and Finkelstein, S. 2013. Reducing unnecessary lab testing in the {ICU} with artificial intelligence. *IntL. J. of Medical Informatics* 82(5):345 – 358.

Fialho, A.; Cismondi, F.; Vieira, S.; Reti, S.; Sousa, J.; and Finkelstein, S. 2012. Data mining using clinical physiology at discharge to predict {ICU} readmissions. *Exp. Sys. with App.* 39(18):13158–65.

Herland, M.; Khoshgoftaar, T. M.; and Wald, R. 2013. Survey of clinical data mining applications on big data in health informatics. In *Proc. ICMLA '13 - Volume 02*, 465–472.

Mao, Y.; Chen, W.; Chen, Y.; Lu, C.; Kollef, M.; and Bailey, T. An integrated data mining approach to real-time clinical monitoring and deterioration warning. In *Proc. SIGKDD '12*, 1140–1148.

Physionet-MIMICIII. https://www.physionet.org/physiobank/database/mimic3cdb/.

Pirracchio, R.; Petersen, M. L.; Carone, M.; Rigon, M. R.; Chevret, S.; and van der Laan, M. J. 2015. Mortality prediction in intensive care units with the Super ICU Learner Algorithm (SICULA): a population-based study. *Lancet Respir Med* 3(1):42–52.

Pontes, B.; Girldez, R.; and Aguilar-Ruiz, J. S. 2015. Biclustering on expression data: A review. *Journal of Biomedical Informatics* 57(Supplement C):163 – 180.

Weka. www.cs.waikato.ac.nz/ml/weka/.